AF IFW

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of:

Ajay Kumar, et al.

Serial No. 10/087,197

Filed: March 1, 2002

For: Incremental Saves for Efficient Distributed State Stores

Group Art Unit: 2166

Examiner: Hwang, Joon H.

Atty. Dkt. No.: 5681-11800
P7519

CERTIFICATE OF MAILING
37 C.F.R. § 1.8

I hereby certify that this correspondence is being deposited with the U.S. Postal Service with sufficient postage as First Class Mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on the date indicated below:

Robert C. Kowert
Name of Registered Representative

September 25, 2006
Date

Signature

APPEAL BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir/Madam:

Further to the Notice of Appeal filed July 24, 2006, Appellants present this Appeal Brief. Appellants respectfully request that the Board of Patent Appeals and Interferences consider this appeal.

I. REAL PARTY IN INTEREST

The subject application is owned by Sun Microsystems, Inc., a corporation organized and existing under and by virtue of the laws of the State of Delaware, and now having its principal place of business at 4150 Network Circle, Santa Clara, CA 95054.

II. RELATED APPEALS AND INTERFERENCES

No other appeals, interferences or judicial proceedings are known which would be related to, directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

III. STATUS OF CLAIMS

Claims 1, 3-9, 11-15 and 17-20 are pending and stand finally rejected. Claims 2, 10, and 16 were previously cancelled. The rejection of claims 1, 3-9, 11-15, and 17-20 is being appealed. A copy of the appealed claims, as currently pending, is included in the Claims Appendix herein below.

IV. STATUS OF AMENDMENTS

No amendments to the claims have been submitted subsequent to the final rejection.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Independent claim 1 is directed toward a system including a distributed store and a first application server of a plurality of application servers (*See e.g.,* Fig. 1, application server 104 and distributed store 110; page 7, lines 9-10 and lines 21-22). The distributed store includes a primary state of session data configured for access by the plurality of application servers (*See e.g.,* Fig. 1, primary state 112; page 7, lines 21-22). Additionally, the session data comprises a plurality of attributes (*See e.g.,* Fig. 8; page 13, lines 27-29; page 16, lines 4-17). The first application server of the plurality of application servers includes a client state of the session data accessible to processes executing within the application server (*See e.g.,* Fig. 1, process 106 and client state 108; page 7, lines 19-21). The first application server is configured to track accesses of the individual attributes of the client state (*See e.g.,* Fig. 8; page 16, lines 21-22). To track accesses of the individual attributes of the client state, the first application server is configured to store information identifying the accessed individual attributes (*See e.g.,* Fig. 8; page 16, lines 21-22). Furthermore, the distributed store is configured to synchronize the primary state with the client state according to the tracked accessed individual attributes (*See e.g.,* page 4, lines 20-21; page 16, lines 10-12).

Independent claim 8 is directed toward a system including a distributed store means, a first application server of a plurality of application servers, and means for tracking accesses of one or more attributes in the client state coupled to or within the first application server (*See e.g.,* Fig. 1, Fig. 8, application server 104 and distributed store 110; page 7, lines 9-10 and lines 21-22; page 16, lines 21-22). The distributed store means includes a primary state of session data configured for access by a plurality of application servers (*See e.g.,* Fig. 1, primary state 112; page 7, lines 21-22). The session data includes one or more individual attributes (*See e.g.,* Fig. 8; page 13, lines 27-29; page 16, lines 4-17). The first application server of the plurality of application servers includes a client state of the session data (*See e.g.,* Fig. 1, process 106 and client state 108; page 7, lines 19-21). The means for tracking accesses of the individual attributes in the client state includes means for storing information identifying the accessed individual attributes (*See e.g.*, Fig. 8; page 16, lines 21-22). The system also includes means for synchronizing the primary state with the client state according to the provided accessed individual attributes (*See e.g.*, page 4, lines 20-21; page 16, lines 10-12).

Independent claim 9 is directed toward a method of tracking accesses of individual attributes in a client state of session data on a first application server and synchronizing a primary state of the session data with the client state (*See e.g.*, Fig. 8; page 4, lines 20-21; page 16, lines 10-12 and 21-22). The session data is accessible to one or more processes executing within the application server (*See e.g.*, Fig. 1; page 7, lines 21-22). Also, the tracking includes the first application server maintaining information identifying the accessed individual attributes (*See e.g.*, Fig. 8; page 16, lines 21-22). Additionally, the synchronizing of the primary state of the session data with the client state includes applying the tracked accessed individual attributes to the session data of the primary state (*See e.g.*, page 4, lines 20-21; page 16, lines 10-12). The primary state is configured for access by a plurality of application servers including the first application server (*See e.g.*, Fig. 2; page 9, line 27 - page 10, line 2).

Independent claim 15 is directed toward a tangible computer accessible storage medium including software instructions computer-executable to implement tracking accesses of individual attributes in a client state of session data on a first application server and synchronizing a primary state of the session data with the client state (*See e.g.*, Fig. 8; page 4, lines 20-21; page 16, lines 10-12 and 21-22). The session data is accessible to one or more processes executing within the application server (*See e.g.*, Fig. 1; 112; page 7, lines 21-22). The tracking includes the first application server maintaining information identifying the access individual attributes (*See e.g.*, Fig. 8; page 16, lines 21-22). Also the synchronizing of the primary state of the session data with the client state includes applying the tracked accessed individual attributes to the session data of the primary state (*See e.g.*, page 4, lines 20-21; page 16, lines 10-12). Furthermore, the primary state is configured for access by a plurality of application servers including the first application server (*See e.g.*, Fig. 2; page 9, line 27 - page 10, line 2).

The summary above describes various examples and embodiments of the claimed subject matter; however, the claims are not necessarily limited to any of these examples and embodiments. The claims should be interpreted based on the wording of the respective claims.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Claims 1, 3-5, 8, 9, 11, 12, 15, 17 and 18 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Montero et al. (U.S. Publication 2002/0143958) (hereinafter "Montero") in view of Bauer et al. (U.S. Patent 5,870,759) (hereinafter "Bauer").

2. Claims 6, 13 and 19 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Montero in view of Bauer and further in view of Morris (U.S. Patent 5,813,017).

3. Claims 7, 14 and 20 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Montero in view of Bauer and further in view of Lin et al. (U.S. Patent 6,546,135) (hereinafter "Lin").

VII. ARGUMENT

First Ground of Rejection

Claims 1, 3-5, 8, 9, 11, 12, 15, 17 and 18 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Montero in view of Bauer. Appellants traverse this rejection for at least the following reasons. Different groups of claims are addressed under their respective subheadings.

Claims 1, 8, 9, 11, 15 and 17

Regarding claim 1, contrary to the Examiner's assertion, Montero in view of Bauer fails to teach or suggest a first application server of the plurality of application servers, comprising a client state of the session data accessible to processes executing within the application server, wherein the first application server is configured to track accesses of the individual attributes of the client state, wherein to track accesses of the individual attributes of the client state, the first application server is configured to store information identifying the accessed individual attributes. Montero teaches a system in which a central database of session data is updated with session data modified in individual application servers periodically, such as according to a certain time interval or after a certain number of changes (Montero, Abstract and paragraphs [0026] and [0049]). Montero does not teach claim 1 as recited above. The Examiner, regarding the rejection of claim 1, cites paragraphs [0011], [0014], [0020], [0026], [0035], [0036], [0042] and [0046] of Montero. However, these paragraphs do not mention *an application server configured to track accesses of the individual attributes of the client state, wherein to track accesses of the individual attributes of the client state, the first application server is configured to store information identifying the accessed individual attributes.* Instead, paragraphs [0011] and [0014] describe how previous methods utilize cookies for servicing session requests, and paragraph [0020] describes a previous method that updates the session data each time the data is changed. Paragraph [0026] summarizes Montero's system including a description of how application servers write copies of session data to common shared databases at designated, periodic times, or after a specified number of changes to session data have been made. Paragraphs [0035] and [0036] describes the purpose of application servers in a server farm with an associated database server for storing backup data of the client sessions. Paragraph [0042] describes how the use of a load balance scheme may ensure that http requests in a particular session are always sent to the same application server. Finally, paragraph [0046] merely mentions that if enough memory is not provided to hold all of the active session data, creation of a new session may result in the need to write out the oldest locally stored http session to the shared database.

Thus, none of the Examiner's cited passages mentions *an application server configured to track accesses of the individual attributes of the client state, wherein to track accesses of the individual attributes of the client state, the first application server is configured to store information identifying the accessed individual attributes*. In fact, nowhere does Montero these features of claim 1. In contrast, Montero teaches only that application servers save their copies of session data to the shared session database periodically to reduce the number of writes to the database, thereby reducing strain on system resources. For instance, Montero teaches, "instead of updating the session data in the database after every request or every attribute change, each of the servers maintains *a fully current copy* of the http session data in its local RAM, but writes a *copy* of the session data to the central database only at specified intervals" (emphasis added, Montero, paragraph [0039]). **Since Montero specifically teaches away from updating the session data in the database after every request or every attribute change, there would be no reason *to store information identifying the accessed individual attributes* to track individual attribute accesses in Montero.** Thus, it would not make sense to modify Montero according to the teachings of another reference to store information identifying the accessed individual attributes to track individual attribute accesses.

In the Advisory Action the Examiner argues, "Montero also teaches updating session data in the database after a specified number of changes to the session data have been made; therefore, if x is the specified number, then x-1 updates are held by the application server [until] there are x number of updates". However, Appellants' argument is not that Montero fails to maintain the number of updates to the session data. Instead, Appellants' argument is that Montero teaches away from an application server configured to track accesses of the individual attributes of the client state, wherein to track accesses of the individual attributes of the client state, the first application server is configured to store information identifying the accessed individual attributes. **Counting the number of times session data is updated and then writing the entire copy of the session data after x number of updates, as argued by the Examiner, not only fails to teach or suggest, but also teaches away from, storing information *identifying* the accessed *individual* attributes to track accesses of individual attributes of the client state.**

As noted above, Montero's application servers do not track accesses of the individual attributes of the client state by storing information identifying the accessed individual attributes. Instead, an application server in Montero's system either keeps track of how long it has been since its respective copy of the session data has been copied to the session database or keeps track of the number of times its copy of the session data has been changed or both. Rather than keep track of the *individual accesses of particular attributes by storing information identifying the accessed individual attributes*, Montero refers to the number of times a local copy of session data has been changed. For example, Montero describes how a local copy of the session data may be copied to the session database "after 3 updates to the local copy of the session data" (Montero, paragraph [0049]). Montero additionally teaches comparing a "last access time" to the current time to determine whether a local copy data has been modified (Montero, paragraph [0053]). Comparing the numbers of updates or last write times of the session data to specified values is not *tracking individual accesses of the attributes of the client state, wherein to track the individual accesses, the first application server is configured to store information identifying the accessed individual attributes*. Moreover, Montero's system looks at updates to the session data as a whole, but clearly does not identify individual accessed attributes of the session data. Furthermore, as noted above, it would not make sense in Montero's system to identify particular accessed attributes of the session data since Montero specifically writing a copy of the session data to the central database "only at specified intervals" "instead of updating the session data in the database after every request or every attribute change". Thus, Montero fails to teach or suggest *an application server configured to track accesses of the individual attributes of the client state, wherein to track accesses of the individual attributes of the client state, the first application server is configured to store information identifying the accessed individual attributes.*

In the Final Action on p. 6, the Examiner agreed that Montero does not disclose that *the first application server is configured to store information identifying the accessed individual attributes to track accesses of the individual attributes of the client state.* The Examiner instead relies on Bauer. However, Bauer teaches that each of the clients (which are not application servers) accesses its own client database, which may later be synchronized with the single central database by the database synchronizer. Appellants note that a database as described in Bauer refers to a collection of data that is manipulated by clients. More specifically, Bauer's data is described as tabular data for order information, delivery status, or field service information (1:4-14), which may be manipulated by clients in a database. The order information, delivery status, or field service information stored in the database in Bauer is not *session data.* Session data is a well-understood concept in the art of application servers, and the data stored in the databases in Bauer is clearly not described as session data. Bauer does not pertain to the field of session data as used by application servers. Thus, the Examiner's reliance on Bauer to teach an <u>application server</u> configured to store information identifying the accessed individual attributes of <u>session data</u> is clearly improper.

Since Bauer has nothing to do with application servers or session data, there would be no reason to apply its teachings to Montero, especially in light of the fact that Montero specifically teaches writing a copy of the session data to the central database "only at specified intervals" "instead of updating the session data in the database after every request or every attribute change". Thus, for at least the reasons provided above, Montero in view of Bauer fails to teach or suggest *wherein to track accesses of the* *individual attributes of the client state, the first application server is configured to store information identifying the accessed individual attributes.*

In the Response to Arguments section of the Final Action, the Examiner contends that Appellants' previous arguments were against the cited referenced individually. To the contrary, Appellants' arguments above show that Montero and Bauer, whether considered alone or in combination, fail to teach or suggest that the first application server is configured to track accesses of the individual attributes of the client state, wherein to track accesses of the individual attributes of the client state, the first application server is configured to store information identifying the accessed individual attributes, as recited in claim 1. Moreover, any statement regarding an individual reference is to show that the Examiner's reliance on the respective reference is misplaced.

Furthermore, the Examiner has failed to provide a proper motivation for modifying the system of Montero. The Examiner states that, "it would have been obvious to one having ordinary skill in the art at the time the invention was made to utilize the teaching of Bauer to the system of Montero in order to **reduce communication costs and delays in data synchronization**" (emphasis added). As stated by the Examiner, and as disclosed in Bauer, column 1, lines 55-60, and elsewhere, Bauer's system seeks to **minimize delays in synchronization of modified client data with a database storing data that is not fully current**; however, Montero **purposefully introduces delays by only updating at specified time intervals**. More specifically, as noted above, Montero teaches that application servers save their fully-current copies of session data to the shared session database periodically in order to reduce the number of writes to the database, thereby reducing strain on system resources. For instance, Montero teaches, "instead of updating the session data in the database after every request or every attribute change, each of the servers maintains a fully current copy of the http session data in its local RAM, but writes a copy of the session data to the central database only at specified intervals" (emphasis added, Montero, paragraph [0039]). Thus, Montero reduces the number of writes to the database by purposefully introducing delays in the form of periodic writes, and maintains a fully current copy of the session data while Bauer, on the other hand, minimizes delay in synchronization of data between the client and the database, where neither the client nor the database maintain a fully current copy of the tabular data. Therefore, Montero and Bauer clearly teach away from their combination. It is improper to combine references where the references teach away from their combination. *In re Grasselli*, 218 USPQ 769, 779 (Fed. Cir. 1983). It is not proper to pick and chose isolated teachings from a reference. Instead, the Examiner must consider each reference as a whole; it is improper to combine references where the references **teach away** from their combination. M.P.E.P. § 2141.02, last paragraph; *W.L. Gore & Associates, Inc. v. Garlock, Inc.*, 721 F.2d 1540, 220 USPQ 303 (Fed. Cir. 1983), *cert. denied*, 469 U.S. 851 (1984).

In the Response to Arguments section of the Final Action, the Examiner points out that motivation to combine reference may come from the references themselves, from knowledge of those skilled in the art or from the nature of the problem to be solved. The Examiner provides a summary of the features of both Montero and Bauer relied on by the Examiner. However, the Examiner has failed to provide any rebuttal to Appellants' argument regarding the fact that Montero and Bauer clearly teach away from their combination, as discussed above. The fact that a motivation to combine reference may come from the references themselves, from knowledge of those skilled in the art or from the nature of the problem to be solved does not overcome the fact that the specific references cited by the Examiner teach away from their combination.

Furthermore, even were Montero and Bauer combinable, which Appellants argue they are not, the suggested combination would not provide all the features recited in claim 1, as described above. As noted in the M.P.E.P 2143.03, "[t]o establish *prima facie* obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art." *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974). As shown above, neither reference, whether considered alone or in combination, teaches or suggests a first application server of the plurality of application servers, comprising a client state of the session data accessible to processes executing within the application server, wherein the first application server is configured to track accesses of the individual attributes of the client state, wherein to track accesses of the individual attributes of the client state, the first application server is configured to store information identifying the accessed individual attributes.

Thus, for at least the reasons above, the rejection of claim 1 is not supported by the cited art and removal thereof is respectfully requested. Claims 8, 9, and 15 recite language similar to that of claim 1; therefore, the arguments above apply equally to those claims, as well.

Claim 3

Regarding claim 3, Montero in view of Bauer fails to teach or suggest that *the first application server is further configured to track mutable individual attributes and not track immutable individual attributes.* As discussed above in regard to claim 1, the cited art does not teach or suggest tracking an application server tracking accesses of individual attributed of session data, let alone tracking mutable individual attributes and not tracking immutable individual attributes. The Examiner relies on Bauer, citing col. 3, lines 53-60 and col. 28, lines 28-52, where Bauer describes that a developer may specify which fields are not modifiable at a client to enable Bauer's database synchronizer to "omit such fields in the before-image table as long as the field are also not used to uniquely identify a given row." However, as noted above regarding the rejection of claim 1, Since Montero specifically teaches away from updating the session data in the database after every request or every attribute change, there would be no reason to track individual attribute accesses in Montero. Similarly, there would be no reason to track mutable individual attributes and not track immutable individual attributes in Montero. Moreover, it would not make sense to modify Montero according to the teachings of another reference to track mutable individual attributes and not track immutable individual attributes. Furthermore, even were Montero and Bauer combinable, which Appellants argue they are not, as argued above, the suggested combination would not provide all the features recited in claim 3, as described above.

Thus, for at least the reasons above, the rejection of claim 3 is not supported by the cited art and removal thereof is respectfully requested.

Claim 4

Regarding claim 4, Montero in view of Bauer fails to teach or suggest that to synchronize the primary state with the client state, the distributed store is further configured to synchronize only mutable individual attributes. In the rejection of claim 1, the Examiner equates the distributed store with the "common session database" of Montero. However, Montero clearly fails to teach or suggest the limitation *to synchronize the primary state with the client state, the distributed store is further configured to synchronize only mutable individual attributes.* As described in the rejection of claim 4, the Examiner rejects claim 4 on the basis of the rejection of claim 3. Therefore, the Examiner relies on Bauer, citing col. 3, lines 53-60 and col. 28, lines 28-52, where Bauer describes that a developer may specify which fields are not modifiable at a client to enable Bauer's database synchronizer to "omit such fields in the before-image table as long as the field are also not used to uniquely identify a given row." However, as noted above regarding the rejection of claim 1, since Montero specifically teaches away from updating the session data in the database after every request or every attribute change, there would be no reason to synchronize individual attributes in Montero. Similarly, there would be no reason to synchronize only mutable individual attributes in Montero. Moreover, it would not make sense to modify Montero according to the teachings of another reference to synchronize only mutable individual attributes. Furthermore, even were Montero and Bauer combinable, which Appellants argue they are not, as argued above, the suggested combination would not provide all the features recited in claim 4, as described above.

Thus, for at least the reasons above, the rejection of claim 4 is not supported by the cited art and removal thereof is respectfully requested.

Claims 5, 12 and 18

Regarding claim 5, the Examiner asserts that "Montero teaches the distributed store is configured to update the primary state with the subset of the accessed attributes that have been modified for synchronizing the primary state with the client state" in paragraph [0026]. However, paragraph [0026] summarizes Montero's system including a description of how application servers write copies of session data to common shared databases at designated, periodic times, or after a specified number of changes to session data have been made. The cited art does mention anything about *the distributed store is configured to update the primary state with the* ***subset of the accessed individual attributes that have been modified***. Contrary to the Examiner's assertion, Montero clearly fails to teach or suggest this claim limitation.

Furthermore, the Examiner admits that Montero does not disclose performing a comparison of the tracked accessed attributes and a benchmark of the session data comprising a previous version of the one or more attributes. Thus, the Examiner relies on Bauer to disclose this claim limitation. However, as noted above regarding the rejection of claim 1, since Montero specifically teaches away from updating the session data in the database after every request or every attribute change, there would be no reason to perform a comparison of the tracked accessed individual attributes and a benchmark of the session data comprising a previous version of the one or more individual attributes to determine a subset of the tracked accessed individual attributes that are modified in respect to the benchmark of the session data in Montero. Moreover, it would not make sense to modify Montero according to the teachings of another reference to determine a subset of the tracked accessed individual attributes. Furthermore, even were Montero and Bauer combinable, which Appellants argue they are not, as argued above, the suggested combination would not provide all the features recited in claim 5, as described above.

Thus, for at least the reasons above, the rejection of claim 5 is not supported by the cited art and removal thereof is respectfully requested. Claims 12 and 18 recite language similar to that of claim 5; therefore, the arguments above apply equally to those claims, as well.

Second Ground of Rejection

Claims 6, 13 and 19 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Montero in view of Bauer and further in view of Morris. Appellants traverse this rejection for at least the following reasons. Different groups of claims are addressed under their respective subheadings.

Claims 6, 13, 19

Regarding claim 6, Montero in view of Bauer in further view of Morris fails to disclose *wherein, in said comparison, the first application server is configured to perform a binary comparison of the tracked accessed individual attributes and the benchmark of the session data to determine a subset of the tracked accessed individual attributes that are modified in respect to the benchmark of the session data.* The Examiner admits that Montero in view of Bauer fails to disclose a binary comparison. Thus, the Examiner relies on Morris to disclose this claim limitation. The Examiner cites the abstract and column 11, line 47 through column 12, line 13. However, these citations describe a method of comparing a compressed file to a base file by compressing the base file and performing a binary comparison of the compressed file and the compressed base file. Morris fails to disclose *wherein, in said comparison, the first application server is configured to perform* ***a binary comparison of the tracked accessed individual attributes and the benchmark of the session data to determine a subset of the tracked accessed individual attributes that are modified in respect to the benchmark of the session data.***

Furthermore, the Examiner asserts "[t]herefore, based on Montero in view of Bauer, and further in view of Morris, it would have been obvious to one having ordinary skill in the art at the time the invention was made to utilize the teachings of Morris to the system of Montero **in order to determine differences of two versions of data**, thereby performing an effective database synchronization" (emphasis added). Appellants respectfully submit that no teaching, motivation, or suggestion exists to combine the teachings of Morris with the teachings of Montero. Clearly, one with ordinary skill in the art, based on Montero in view of Bauer and further in view of Morris, would not be motivated to combine the teachings of Morris and Montero to "determine differences of two versions of data." Even if Montero and Bauer were combinable, which Appellants argue they are not, as argued above, one with ordinary skill in the art would simply use the teachings of Bauer to "determine differences of two versions of data". However, as noted above regarding the rejection of claim 1, Montero specifically teaches away from updating the session data in the database after every request or every attribute change; therefore, there would be no reason to perform a binary comparison of the tracked accessed individual attributes and the benchmark of the session data to determine a subset of the tracked accessed individual attributes that are modified in respect to the benchmark of the session data in Montero. Moreover, it would not make sense to modify Montero according to the teachings of another reference to determine a subset of the tracked accessed individual attributes. Additionally, the prior art references lack a teaching or suggestion to combine the teachings of Montero and the teachings of Morris. As the Examiner is certainly aware, it is impermissible to reconstruct the claimed invention from selected pieces of prior art absent of some suggestion, teaching, or motivation in the prior art to do so. Furthermore, even were Montero and Morris combinable, which Appellants argue they are not, as argued above, the suggested combination would not provide all the features recited in claim 6, as described above.

Thus, for at least the reasons described above, the rejection of claim 6 is not supported by the prior art and removal thereof is respectfully requested. Claims 13 and 19 recite language similar to that of claim 6; therefore, the arguments above apply equally to those claims, as well.

Third Ground of Rejection

Claims 7, 14 and 20 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Montero in view of Bauer and further in view of Lin. Appellants traverse this rejection for at least the following reasons. Different groups of claims are addressed under their respective subheadings.

Claims 7, 14 and 20

Regarding claim 7, Montero in view of Bauer in further view of Lin fails to disclose *wherein, in said comparison, the first application server is configured to perform an object graph comparison of the tracked accessed individual attributes and the benchmark of the session data to determine a subset of the tracked accessed individual attributes that are modified in respect to the benchmark of the session data.* The Examiner admits that Montero and Bauer do not disclose an object graph comparison. Therefore, the Examiner relies on Lin to disclose this claim limitation. The Examiner cites the abstract; column 7, line 40 through column 8, line 14; and Figure 5; however, these citations do not disclose *wherein, in said comparison, the first application server is configured to perform an object graph comparison of the tracked accessed individual attributes and the benchmark of the session data to determine a subset of the tracked accessed individual attributes that are modified in respect to the benchmark of the session data.* Moreover, the cited art discloses a method of <u>comparing **multimedia content** to produce a **similarity score**</u>. As described above, session data is a well-understood concept in the art of application servers, and the data compared in Lin is clearly not described as session data. Lin does not pertain to the field of session data as used by application servers. Furthermore, Lin fails to teach or suggest the comparison of **tracked accessed individual attributes** and **benchmark session data**. Thus, the Examiner's reliance on Lin to teach an application server that is configured to perform an <u>object graph comparison of the tracked accessed individual attributes and the benchmark of the session data to determine a subset of the tracked accessed individual attributes</u> is clearly improper.

Furthermore, the Examiner asserts "[t]herefore, based on Montero in view of Bauer, and further in view of Lin, it would have been obvious to one having ordinary skill in the art at the time the invention was made to utilize the teachings of Lin to the system of Montero **in order to determine differences of two version of data**, thereby performing an effective database synchronization" (emphasis added). Appellants respectfully submit that no teaching, motivation, or suggestion exists to combine the teachings of Lin with the teachings of Montero. Clearly, one with ordinary skill in the art, based on Montero in view of Bauer and further in view of Lin, would not be motivated to combine the teachings of Lin and Montero to "determine differences of two versions of data." Even if Montero and Bauer were combinable, which Appellants argue they are not, as argued above, one with ordinary skill in the art would simply use the teachings of Bauer to "determine differences of two versions of data". However, as noted above regarding the rejection of claim 1, Montero specifically teaches away from updating the session data in the database after every request or every attribute change; therefore, there would be no reason to perform an object graph comparison of the tracked accessed individual attributes and the benchmark of the session data to determine a subset of the tracked accessed individual attributes that are modified in respect to the benchmark of the session data in Montero. Moreover, it would not make sense to modify Montero according to the teachings of another reference to determine a subset of the tracked accessed individual attributes. Additionally, the prior art references lack a teaching or suggestion to combine the teachings of Montero and the teachings of Lin. As the Examiner is certainly aware, it is impermissible to reconstruct the claimed invention from selected pieces of prior art absent of some suggestion, teaching, or motivation in the prior art to do so. Furthermore, even were Montero and Lin combinable, which Appellants argue they are not, as argued above, the suggested combination would not provide all the features recited in claim 7, as described above.

Thus, for at least the reasons described above, the rejection of claim 7 is not supported by the prior art and removal thereof is respectfully requested. Claims 14 and 20 recite language similar to that of claim 7; therefore, the arguments above apply equally to those claims, as well.

CONCLUSION

For the foregoing reasons, it is submitted that the Examiner's rejection of claims 1, 3-9, 11-15 and 17-20 was erroneous, and reversal of his decision is respectfully requested.

The Commissioner is authorized to charge the appeal brief fee of $500.00 and any other fees that may be due to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5681-11800/RCK. This Appeal Brief is submitted with a return receipt postcard.

Respectfully submitted,

Robert C. Kowert
Reg. No. 39,255
Attorney for Appellants

Meyertons, Hood, Kivlin, Kowert & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
(512) 853-8850

Date: September 25, 2006

VIII. CLAIMS APPENDIX

The claims on appeal are as follows.

1. A system, comprising:

a distributed store comprising a primary state of session data configured for access by a plurality of application servers, wherein the session data comprises a plurality of attributes; and

a first application server of the plurality of application servers, comprising a client state of the session data accessible to processes executing within the application server, wherein the first application server is configured to track accesses of the individual attributes of the client state, wherein to track accesses of the individual attributes of the client state, the first application server is configured to store information identifying the accessed individual attributes;

wherein the distributed store is configured to synchronize the primary state with the client state according to the tracked accessed individual attributes.

3. The system as recited in claim 1, wherein, to track accesses of the attributes of the client state, the first application server is further configured to track mutable individual attributes and not track immutable individual attributes.

4. The system as recited in claim 1, wherein, to synchronize the primary state with the client state, the distributed store is further configured to synchronize only mutable individual attributes.

5. The system as recited in claim 1, wherein the first application server is configured to perform a comparison of the tracked accessed individual attributes and a

benchmark of the session data comprising a previous version of the one or more individual attributes to determine a subset of the tracked accessed individual attributes that are modified in respect to the benchmark of the session data; and wherein, to synchronize the primary state with the client state, the distributed store is configured to update the primary state with the subset of the accessed individual attributes that have been modified.

6. The system as recited in claim 5, wherein, in said comparison, the first application server is configured to perform a binary comparison of the tracked accessed individual attributes and the benchmark of the session data to determine a subset of the tracked accessed individual attributes that are modified in respect to the benchmark of the session data.

7. The system as recited in claim 5, wherein, in said comparison, the first application server is configured to perform an object graph comparison of the tracked accessed individual attributes and the benchmark of the session data to determine a subset of the tracked accessed individual attributes that are modified in respect to the benchmark of the session data.

8. A system comprising:

a distributed store means comprising a primary state of session data configured for access by a plurality of application servers, wherein the session data comprises one or more individual attributes;

a first application server of the plurality of application servers comprising a client state of the session data;

means for tracking accesses of the individual attributes in the client state, coupled to or within the first application server, wherein said means for tracking accesses of the individual attributes in the client state comprises means for storing information identifying the accessed individual attributes; and

means for synchronizing the primary state with the client state according to the provided accessed individual attributes.

9. A method comprising:

tracking accesses of individual attributes in a client state of session data on a first application server, wherein the session data is accessible to one or more processes executing within the application server, wherein said tracking comprises the first application server maintaining information identifying the accessed individual attributes; and

synchronizing a primary state of the session data with the client state by applying the tracked accessed individual attributes to the session data of the primary state, wherein the primary state is configured for access by a plurality of application servers including the first application server.

11. The method as recited in claim 9, further comprising storing the primary state in a distributed store accessible to the application servers.

12. The method as recited in claim 9, further comprising determining differences between the tracked accessed individual attributes and a benchmark of the session data comprising a previous version of the one or more individual attributes to detect a subset of the accessed individual attributes that have been modified; wherein, said synchronizing comprises applying only the subset of accessed individual attributes that have been modified to the session data of the primary state.

13. The method as recited in claim 12, wherein said determining differences comprises performing a binary comparison of the tracked accessed individual attributes to the benchmark of the session data.

14. The method as recited in claim 12, wherein said determining differences comprises performing an object graph comparison of the tracked accessed individual attributes to the benchmark of the session data comprising a previous version of the one or more individual attributes.

15. A tangible computer-accessible storage medium, comprising software instructions computer-executable to implement:

tracking accesses of individual attributes in a client state of session data on a first application server, wherein the session data is accessible to one or more processes executing within the application server, wherein said tracking comprises the first application server maintaining information identifying the access individual attributes; and

synchronizing a primary state of the session data with the client state by applying the tracked accessed individual attributes to the session data of the primary state, wherein the primary state is configured for access by a plurality of application servers including the first application server.

17. The tangible computer-accessible storage medium as recited in claim 15, wherein the primary state is comprised in a distributed store.

18. The tangible computer-accessible storage medium as recited in claim 15, wherein the software instructions are further computer-executable to implement determining differences between the tracked accessed individual attributes and a benchmark of the session data comprising a previous version of the one or more individual attributes to detect a subset of the accessed individual attributes that have been

modified; wherein said synchronizing comprises applying only the subset of accessed individual attributes that have been modified to the session data of the primary state.

19. The tangible computer-accessible storage medium as recited in claim 18, wherein said determining differences comprises performing a binary comparison of the tracked accessed individual attributes to the benchmark of the session data comprising a previous version of the one or more individual attributes.

20. The tangible computer-accessible storage medium as recited in claim 18, wherein said determining differences comprises performing an object graph comparison of the tracked accessed individual attributes to the benchmark of the session data comprising a previous version of the one or more individual attributes.

IX. EVIDENCE APPENDIX

No evidence submitted under 37 CFR §§ 1.130, 1.131 or 1.132 or otherwise entered by the Examiner is relied upon in this appeal.

X. RELATED PROCEEDINGS APPENDIX

There are no related proceedings.